Technical white paper

大規模共有メモリーシステムでのGAMESSの利点

はじめに

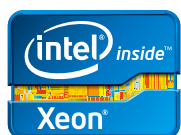
GAMESSはアイオワ州立大学のGordon Groupによって開発された量子力学計算プログラム*1です。Gaussianと異なりライセンスフリーであることから、使用しているユーザも少なくありません。しかしGaussianに比べると若干手間のかかる部分もあり、例えば1つの振動数計算をするのに、構造最適化をしてから本計算をしなければなりません。インテル® Xeon® プロセッサー E7ファミリーを8基（80コア）搭載したクラスタと大規模共有メモリ型サーバに絞り、その特性を調査しました。
計算対象の原子数や分子数によって当然用いられる解法も異なります。このレポートはその全てを網羅する内容ではありませんが、原子数で数十から数百程度のモデルを扱われている皆様にとって何らかの参考になればと思います。

（日本ヒューレット・パッカード（株） プリセールス統括本部 磯田 大典）

2013年4月

まず、GAMESSのドキュメントから対応オプションを転写します（表1）

解析手法	RHF	ROHF	UHF	GVB	MCSCF
SCF Energy	CDFpEP	CDFpEP	CDFpEP	CD-pEP	CDFpEP
SCF Gradient	CDFpEP	CDFpEP	CDFpEP	CD-pEP	CDFpEP
SCF Hessian	CD-p--	CD-p--	-----	CD-p--	-D-p--
MP2 energy	CDFpEP	CDFpEP	CD-pEP	-------	CD-pEP
MP2 gradient	CDFpEP	-D-pEP	CD-pEP	-------	------
CI energy	CDFp--	CD-p--	------	CD-p--	CD-p--
CI gradient	CD-----	-------	-------	-------	------
CC energy	CDFpE-	CDF-E-	-------	-------	------
EOMCC excitations	CD—E-	CD—E-	-------	-------	------
Semi-empirical:					
DFT energy	CDFpEP	CD-pEP	CD-pEP		
DFT gradient	CDFpEP	CD-pEP	CD-pEP		
TD-DFT energy	CDFpEP	------	CD-p--		
TD-DFT gradient	CDFpEP	------	------		
Mopac energy	y	Y	y	y	
Mopac gradient	y	Y	y	n	

表1. 小文字pがparallel対応

GAMESSのプロセス機構

GAMESSのプロセス機構は図1に示される4階層から成り立っています。"chem code" は量子計算の実行カーネル、"DDI code"は並列インターフェイス、"Replicated data" は各プロセスにコピーされる共通データ領域、そして最下層の "Distributed data" は計算ノード全体で必要とされる領域になります。後者2つのプロセスのメモリ量は $SYSTEM文のMWORD, MEMDDIパラメータで指定します。

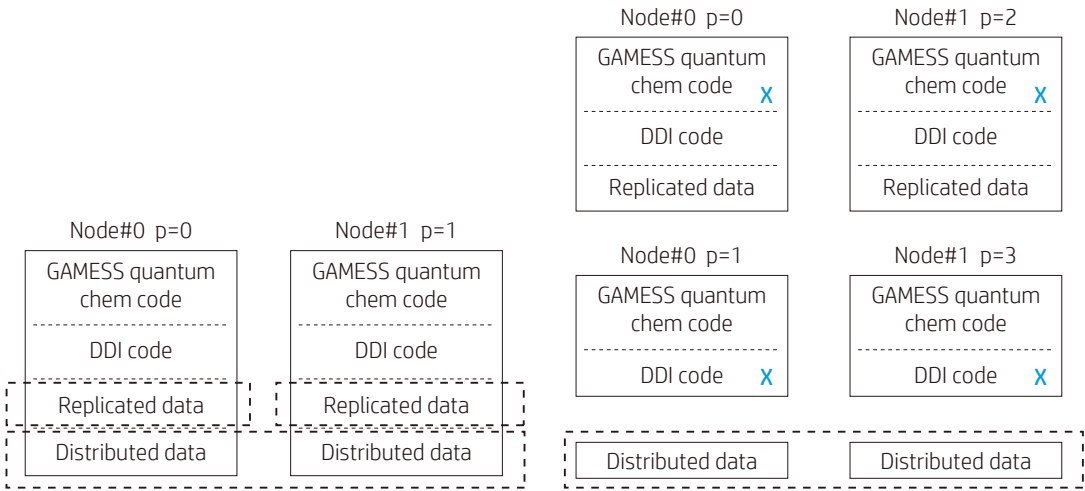



図1. 分散メモリシステムでのGAMESSのプロセス(左は1ノード1コア、右は1ノード2コアの場合)

コアあたりに必要なメモリーは

1 core Memory = MWORD + MEMDDI / (並列数)

で与えられます。MWORD, MEMDDIはword単位で8x10^6を乗じた値です。

図1(右)は各ノードに2プロセスが流れる場合です。最初のプロセスは量子計算を担当する"compute process" (以下CP)、もう一方はデータ転送のみを扱う"data server"(以下DS) と呼ばれるプロセスです。例えば16コア搭載しているサーバでは、それぞれ8つ動くことになり、互いにペアとなります。p=2のプロセスがp=0のデータを取得する場合はp=1に問い合わせを行いp=2に転送されます。p=0はこの通信によって処理が中断される事はありません。この場合もメインプロセスが消費するメモリーは上記と同様ですが、DSが消費するメモリーは数MB程度です。GAMESSではCPとDSが必ず動くことになりますが、共有メモリシステムのDSは殆ど仕事をしません。

以下3つのモデルを用いて考察します。

1) 水分子

並列計算をする必要のない低分子として水分子の基本振動数を求めてみます。これは構造最適化(runtype=optimize)で得られた座標情報を振動計算(runtype=hessian)に渡し、その軌道情報とFCM情報($VEC, $HES)を使って非調和振動数の計算(runtype=VSCF)を行うものです。基底関数はcc-pVDZ、電子相関をMP2/CCSD/ CCSD(T) の3パターンで計算しました。

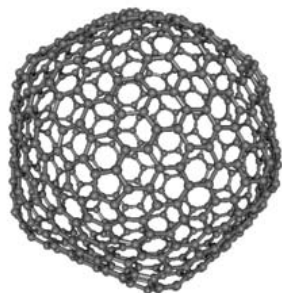
```
$CONTRL SCFTYP=RHF RUNTYP=VSCF CCTYP=CCSD(T) ISPHER=1 $END

 $SYSTEM TIMLIM=100000 MWORDS=100 $END
 $BASIS  GBASIS=CCD $END
 $VSCF NGRID=16 PETYP=DIRECT $END
 $GUESS GUESS=HUCKEL $END
 $DATA
H2O  CCSD(T)/cc-pVDZ Anharmonic Frequency
CNV   2
 O       8.0   0.0000000000   0.0000000000   0.1403674299
 H       1.0  -0.7493341373   0.0000000000  -0.4675002149
 $END
```

	Exp.	MP2	CCSD	CCSD(T)
ν1 (変角)	1595	1610	1629	1622
ν2 (対称振動)	3657	3670	3658	3628
ν3 (逆対称伸縮振動)	3756	3757	3728	3699
(SATA DISK使用)	-	143秒	163秒	131秒
(RAM DISK: /dev/shm使用)	-	77秒	83秒	66秒

表2. 水分子の非調和振動数の計算

各振動数を比較するとMP2が最も実験値に近いと言えます。理由は理論書を読めば分かるかもしれませんが、ここでは割愛させて頂きます。一方、計算負荷は殆ど変わりませんでしたが、低分子であるにも関らずファイルIOは無視できません。HDDとRAM-DISKとの差を比較すると、RAM-DISKの方が凡そ2倍高速になりました。さらに原子数が増えてくると、ディスクIOにかかる割合も無視できなくなりそうです。

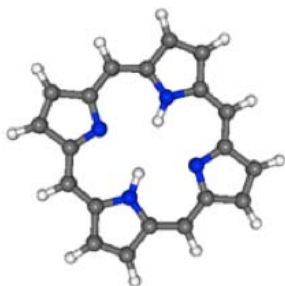
2）カーボンフラーレン C540

フラーレン (fullerene) は、多数の炭素原子のみで構成される、中空な球状のクラスターの総称である。共有結合結晶であるダイヤモンドおよびグラファイトと異なり、数十個の原子からなる構造を単位とする炭素の同素体である

次の例はC540のグラジエント、シングルポイントのエネルギーとその微分の計算になります（PC Gamessのtest2のデータです*2）。基底関数はMOPAC半経験的量子計算法PM3、SCFTYPEはRHF。PM3法は表1にあるとおり並列計算には対応していないため、PM3の性能を上げることは難しいと言えます。このため、GAMESSのビルド方法を変え高次のコンパイルオプションを試してみました。デフォルトオプションに加え、「http://spec.org/cpu2000」*3のコンパイラオプションを例に最適化を施しました。表3の”-xHOST”はCPUの世代によって適用される命令セットであり、SSEやAVX命令を示します。”-ipo”はプロシージャ間の最適化を行うものですが、ビルドに非常に時間を要します。最後にNehelam世代からのTURBOモードを試しました。

```
$CONTRL SCFTYP=RHF RUNTYP=gradient nprint=-5 ISPHER=1 $END
$SYSTEM TIMLIM=3000 MWORDS=100 $END
$BASIS  GBASIS=PM3 $END
$GUESS  GUESS=HUCKEL $END
$scf soscf=.t. $end
$DATA
```

Case	Elapsed Time(sec)	R-PM3 ENERGY	SCF iter.
1 (default)	175	2349.3718701794	15
2 (optimize)	178	2349.3718701796	16
3 (optimize + turbo)	141	2349.3718701796	16

表3. PM3の高速化
Default: -i8 -O2
Optimize: -i8 -xHOST -ipo -O3 -no-prec-div -unroll2 -static -scalar-rep-

3）ポルフィリン C20H16N4

ポルフィリン (porphyrin) は、ピロールが4つ組み合わさって出来た環状構造を持つ有機化合物。分子全体に広がったπ共役系の影響で平面構造をとり、中心部の窒素は鉄やマグネシウムをはじめとする多くの元素と安定な錯体を形成する。

```
$CONTRL RUNTYP=GRADIENT INTTYP=HONDO ICUT=10 CITYP=CIS NPRINT=-5 $END
$SYSTEM TIMLIM=6000 MWORDS=100 $END
$GUESS GUESS=HUCKEL $END
$SCF DIRSCF=.T. $END
$CIS NSTATE=1 CHFSLV=DIIS $END
$DATA
```

GAMESSの並列環境ではMPIの上に独自のライブラリDDIが用いられています。DDIはシステムに依存して動きが異なる為、注意が必要です。今回コンパイル・ビルドした環境を以下に記します。

・GAMESS VERSION = 1 MAY 2012 (R2)

・並列ライブラリ：Intel-MPI 4.1.0

・Mathlib：Intel MKL 13.0

・Fortran：Intel fortran 13.0 [composer_xe_2013.2.146]

テストデータはポルフィリン分子で、PC GAMESSのtest6に相当します。まず最初に表7のクラスタで行いました。この環境でのDDIの動作は図1（右）となります。16コアのうち、CPとDSが各8プロセスづつ動きます。表4および図2は1024コアまで測定した結果です。256コアから並列度の傾きが変わってきています。

Cores	Elapsed Time(sec)	FINAL RHF ENERGY	SCF iter
1	85381	-983.5780915418	24
16	10730	-983.5780915569	16
32	5618	-983.5780915613	23
64	2927	-983.5780915656	25
128	1555	-983.5780915563	21
256	919	-983.5780915667	28
512	662	-983.5780915603	21
1024	444	-983.5780915592	20

表4. ホルフィリンの大規模並列計算(1)

この計算を通じてRHF SCF計算での収束判定が並列数に応じて変わってくることが分かりました。収束判定条件は ”DENSITY MATRIX < 2.00E-05” です。

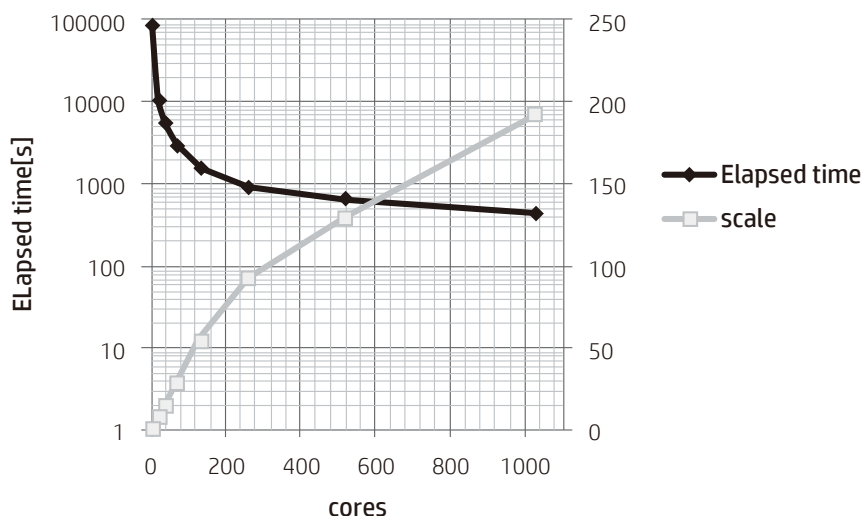



図2. ホルフィリンの大規模並列計算(2)

並列計算の精度は通信の到達順番によって、すなわちMPIランクの算術順番によって変化します。GAMESSではSCF計算の波動関数の収束に至るイタレーション数は実行タイミングや並列数によって変わってきます。デフォルトではMAXIT=30ですが、場合によって30を超えるステップ数を要し、収束しなかった場合には ”UNCONVERGED – MPI ERROR” で異常終了します。次にインターコネクトの影響を調べてみます。このクラスターではInfinibad-FDRとGigabitが接続されています。ジョブに応じて切り替えて使うためにはmpiexecの中で使用するデバイスを明示します。Intel-MPIの場合、I_MPI_DEVICE=「shm or rdssm or ssm」の中から、Infinibandの場合ではrdssmを、Gagabitではssmを使用します。表5に相対比を示しました。SCFの収束数の差異はありますが、256並列までは同等性能、512並列を超えてからその差は顕著になる傾向が分かります。

Cores	SCF iterations infiniband	SCF iterations Gigabit	Elapsed Time Infiniband(rdssm)/Gigabit(ssm)
64	25	21	0.99
128	21	25	0.95
256	28	16	0.97
512	21	19	0.88
1024	20	23	0.67

表5. インターコネクトの違いによる性能差

次に共有メモリシステムでのGAMESSの利点について述べます。GAMESSは分散メモリシステムより共有メモリシステムの方がより優れた性能を発揮すると著者は考えています。

	分散メモリSL230Gen8	共有メモリDL980G7
Processor	Intel E5-2670 2.7Ghz	Intel E7-4870 2.4Ghz
Sockets, Core	2socket, 16core	8socket, 80core
Memory	8GB DIMM x16	32GB DIMM x128
OS	RedHat 6.3	RedHat 6.4
Interconnect	InfiniBand FDR	-

表6. テスト環境

SMPの利点は全てのCPが同一ノード上のMEMDDI領域を参照することで、通信にかかるオーバーヘッドを軽減できる事です。つまり冒頭で触れたMEMDDIで定義される "distributed data" を共有メモリ内に格納できる為です。システム間でMEMDDIのやり取りしていたDSはその役目から開放され、CPUリソースを使う事はありません。その結果CP は2倍のCPUリソースを使うことが出来ます。この設定はIntel-MPIの環境変数 "I_MPI_WAIT_MODE" で制御できます。
(cshの場合) setenv I_MPI_WAIT_MODE enable
とすればよいわけです。またデータ規模に応じてSystemVに準拠するカーネルパラメータを調整する必要があります。これらの値はipcs -lコマンドで参照でき一時的な変更であればsysctl -w kernel.shmmax="xxx" が便利です。

```
kernel.shmmax = 68,719,476,736 (64GB)
kernel.shmall = 4,294,967,296 (4GB)
kernel.shmmni = 4,096 (4GB)
kernel.shm_rmid_forced = 0
vm.hugetlb_shm_group = 0
```

上記の設定の下でDSを起動した場合と起動しない場合の結果を表7-8に示します。使用したシステムは表6のDL980G7になります。

Cores	CP	Elapsed Time	RHF ENERGY	ITERATIONS
8	4	23373	-983.5780915653	15
16	8	11792	-983.5780915632	15
32	16	6202	-983.5780915587	22
64	32	3672	-983.5780915634	15
80	40	3032	-983.5780915579	18

表7. DSを起動した場合

Cores	CP	Elapsed Time	RHF ENERGY	ITERATIONS
8	8	12064	-983.5780915633	15
16	16	6384	-983.5780915652	15
32	32	3590	-983.5780915622	15
64	64	2427	-983.5780915633	15
80	80	2348	-983.5780915652	15

表8. DSを起動しない場合

DSの効果は8並列で2倍、80並列で1.3倍となっています。並列数に応じて差が小さくなる傾向があります。その理由は各プロセス時間のバラツキによるものです。8並列時の各プロセスタイムを見るとそれらは等価ですが、80並列時では最大386秒、全体の計算時間の16%の割合を占めています。この点がDSをoffにしたトレードオフとも言える訳ですが、少なくとも80並列まではオフの効果がある事が分かりました。同様の手法をクラスタシステムで実行すると512並列では500%近く遅くなってしまいました。

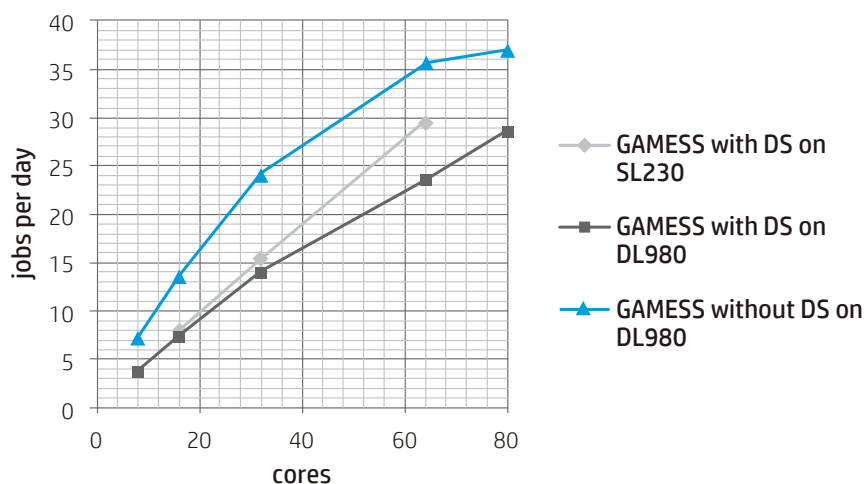



図3. DL980の利点。DSを制御し最大2倍の高速化を実現

最後に摂動法MP2とクラスター展開法CCSDについて考察します。Hatree-Fock法は電子間相互作用を平均場として扱っている為、正確なシュレディンガー方程式の解にはなっていません。厳密解とHFとのずれを電子相関と呼びますが、この電子相関エネルギーを見積方法としてよく使われるのが、MP2やCCSDです。SCF計算においてメモリーは基底関数の2乗に比例します。一方MP2の計算時間は原子数の2乗、CCSDでは5-6乗に比例すると言われています*4。CCSDはMP法よりも計算コストは大きくなりますが、その分精度が高いと言われています。

CCSD, CCSD(T)計算はGradientに対応していない為、ポルフィリンのデータをシングルポイントのエネルギー計算に変更しました。まず図4にCCSD, CCSD(T)の概略を示します。それぞれのGAMESSのプロセスはメモリ上の共有メモリー領域にある "node-replicated" と "distributed storage" を一意に参照します。共有メモリサイズが小さい場合はセグメンテーションエラーを引き起こしますので、shmmaxのパラメータを増やす必要があります。共有メモリシステムの場合には "node-replicated" はこれまでの "process –replicated" とは異なりノード毎に確保される領域です。"distributed storage(MEMDDI)" は各ノードの共有メモリ領域に確保され、DDIの機構に従い、例えばTCP-IPなどで通信されます。表9にEXETYP=CHECKによってメモリサイズを査定した結果を記します。この値は計算可能な最小値であり、実際にはこの値よりも大きな値を設定します。

ここでCCSDの計算時間を予測してみることにしましょう。最終的なCC計算のイタレーション数が不明であるため、MAXCC=1までに要した時間を30倍（MAXCC=30）としました。するとHFの約2000倍の計算量になります。CCSD計算はAOとMOに 処理が分かれますが、AOは並列化される一方、MOは殆どシングルで動きます。そのため、MOの処理時間の短縮が重要になります。全体として更にCPUを追加したとしても効果は限定的であり、アルゴリズムの改変や計算環境の一層の向上が望まれます。

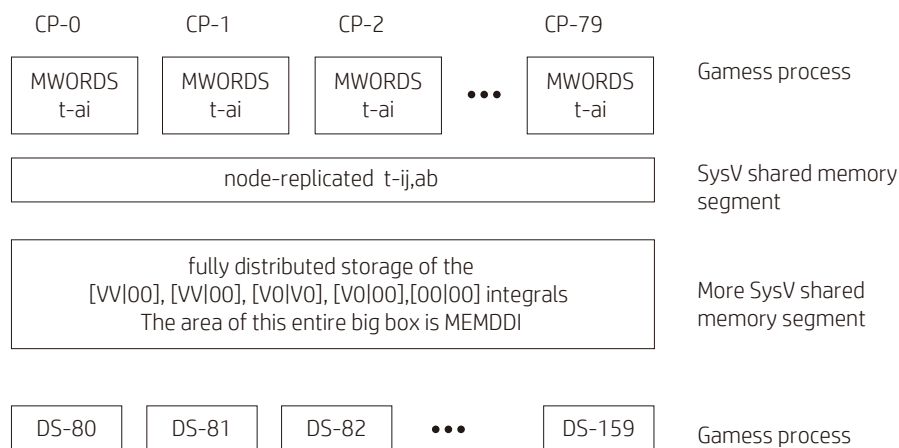



図4. CCSD,CCSD(T)計算のメモリー制御機構

	Replicated memory(MB)	Distributed memory(MB)
HF	36	0.3
MP2	89	4171
CCSD	240	332,152
CCSD(T)	1.2	75,752

表9. EXETYP=CHECKによるメモリ査定

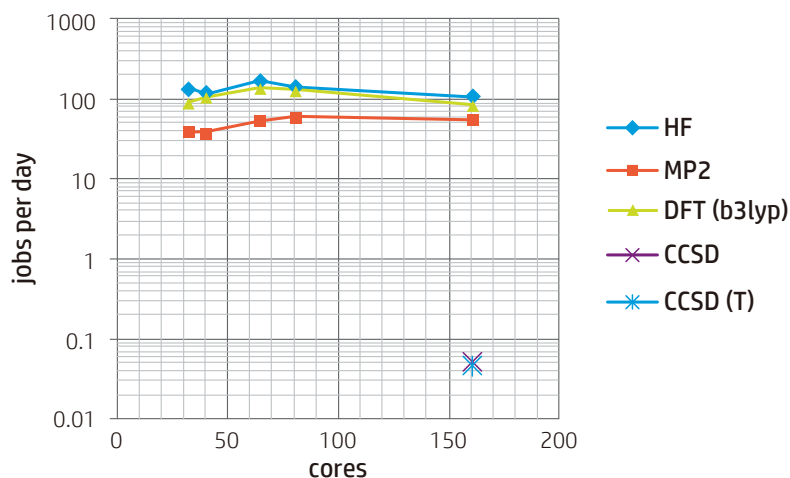



図5. CCSDの計算時間予測（MAXCC=30として算出）

最後に以上を纏めます。

1. メモリ分散システムでは"Data Server"を有効にすることで、ノード間の計算バランスが保たれ、コア数に比例した性能が得られる
2. 共有メモリ型システムでは全てのCPUコアを"Compute Process"に割り当て高速化を図る事が出来る。しかし上限としては80-128並列と推測される
3. MP2は原子数の2乗、CCSDは原子数の4乗に比例します。現在の最速機を活用してもCCSD計算には膨大な時間がかかります
4. 大規模共有メモリ計算機はCCSDのように分散メモリ型では出来ない計算が可能であり、安定した解が得られます

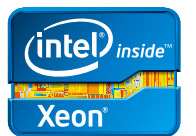
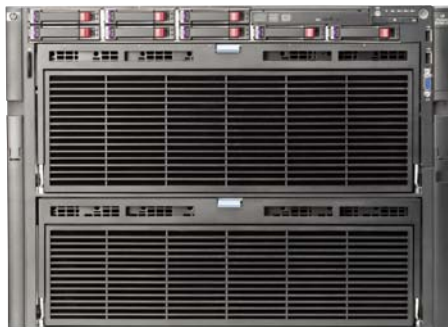
今後は原子数で数百から数千程度のモデルに関するベンチマークとGPGPUライブラリlibcchemの調査を進めて行きたいと思います。

参照：

*1) http://www.msg.ameslab.gov/gamess

*2) http://classic.chem.msu.su/gran/gamess/index.html

*3) http://www.spec.org/cpu2000/

*4) http://www.chem.waseda.ac.jp/nakai/research_dc.html

インテル® Xeon® プロセッサー E7 ファミリー

ハードウェア：**HP ProLiant DL980 G7**
搭載プロセッサー：インテル® Xeon® プロセッサー E7-4870 2.4GHz 8基／80コア
搭載メモリー：32GB DIMM ×128
OS：Red Hat 6.4

安全に関するご注意　ご使用の際は、商品に添付の取扱説明書をよくお読みの上、正しくお使いください。水、湿気、油煙等の多い場所に設置しないでください。火災、故障、感電などの原因となることがあります。

お問い合わせはカスタマー・インフォメーションセンターへ

03-5749-8328 月～金 9:00～19:00　土 10:00～17:00(日、祝祭日、年末年始および5/1を除く)

機器のお見積もりについては、代理店、または弊社営業にご相談ください。

HP ProLiantに関する情報は http://www.hp.com/jp/proliant

Intel、インテル、Intel ロゴ、Intel Inside、Intel Inside ロゴ、Xeon、Xeon Insideは、アメリカ合衆国および／
またはその他の国におけるIntel Corporationの商標です。
記載されている会社名および商品名は、各社の商標または登録商標です。
記載事項は2013年3月現在のものです。
本カタログに記載されている情報は取材時におけるものであり、閲覧される時点で変更されている可能性があります。あらかじめご了承ください。

日本ヒューレット・パッカード株式会社
〒136-8711 東京都江東区大島2-2-1